



语法分析

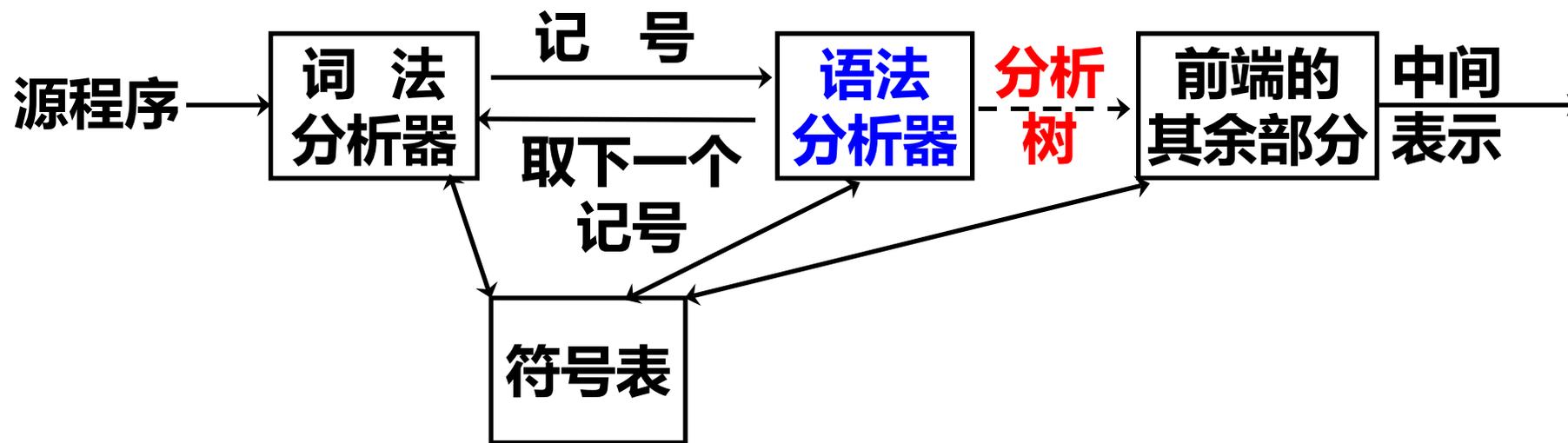
自底向上-移进规约

李诚

国家高性能计算中心(合肥)、信息与计算机国家级实验教学示范中心

计算机科学与技术学院

2024年09月18日



• 自底向上分析方法

- 归约(右推导的逆过程)
- 句柄(可归约串), 可能不唯一
- 移进-归约分析方法
- 冲突: 移进-归约、归约-归约



• 自顶向下 (Top-down)

- 针对输入串，从文法的开始符号出发，尝试根据产生式规则 **推导 (derive)** 出该输入串。
- 即便是进行消除左递归、提取左公因子操作，仍然存在一些程序语言，他们对应的文法不是LL(1)

• 自底向上 (Bottom-up)

- 针对输入串，尝试根据产生式规则 **归约 (reduce)** 到文法的开始符号。
- 比top-down分析方法更一般化



归约(Reduce)



- 每一步，**特定子串**被替换为相匹配的某个**产生式左部的非终结符**
- 最终，把**输入串**归约成文法的**开始符号**



归约(Reduce)



- 每一步，**特定子串**被替换为相匹配的某个**产生式左部的非终结符**
- 最终，把**输入串**归约成文法的**开始符号**

例 $S \rightarrow aABe$

$A \rightarrow Abc / b$

$B \rightarrow d$

输入串: $abbcde$



归约(Reduce)



- 每一步，**特定子串**被替换为相匹配的某个**产生式左部的非终结符**
- 最终，把**输入串**归约成文法的**开始符号**

例 $S \rightarrow aABe$

$A \rightarrow Abc / b$

$B \rightarrow d$

输入串: $abbcde$

ab (读入 ab)

a

b



归约(Reduce)



- 每一步，**特定子串**被替换为相匹配的某个**产生式左部的非终结符**
- 最终，把**输入串**归约成文法的**开始符号**

例 $S \rightarrow aABe$

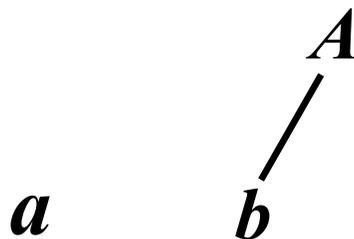
$A \rightarrow Abc / b$

$B \rightarrow d$

输入串: $abbcde$

ab (读入 ab)

aA (归约)





归约(Reduce)



- 每一步，**特定子串**被替换为相匹配的某个**产生式左部的非终结符**
- 最终，把**输入串**归约成文法的**开始符号**

例 $S \rightarrow aABe$

$A \rightarrow Abc / b$

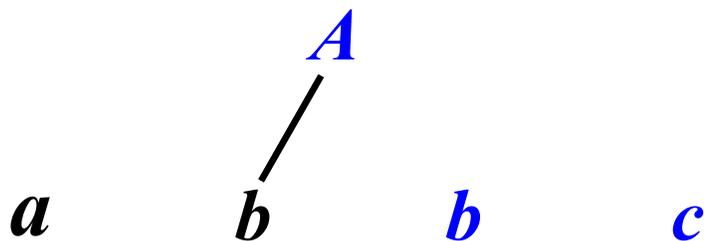
$B \rightarrow d$

输入串: $abbcde$

ab (读入 ab)

aA (归约)

$aAbc$ (再读入 bc)





归约(Reduce)



- 每一步，**特定子串**被替换为相匹配的某个**产生式左部的非终结符**
- 最终，把**输入串**归约成文法的**开始符号**

例 $S \rightarrow aABe$

$A \rightarrow Abc / b$

$B \rightarrow d$

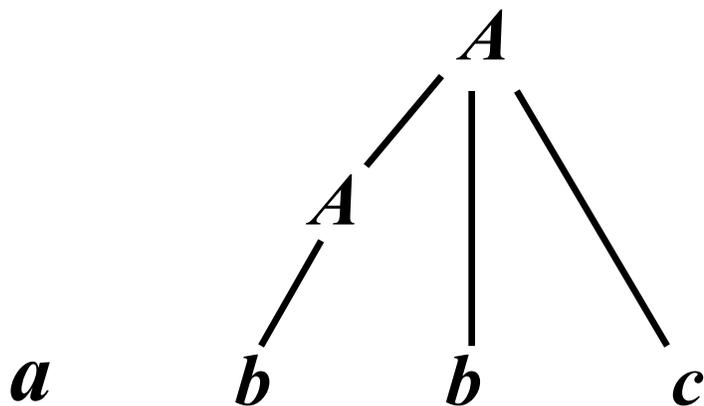
输入串: $abbcde$

ab (读入 ab)

aA (归约)

$aAbc$ (再读入 bc)

aA (归约)





归约(Reduce)



- 每一步，特定子串被替换为相匹配的某个产生式左部的非终结符
- 最终，把输入串归约成文法的开始符号

例 $S \rightarrow aABe$

$A \rightarrow Abc / b$

$B \rightarrow d$

输入串: $abbcde$

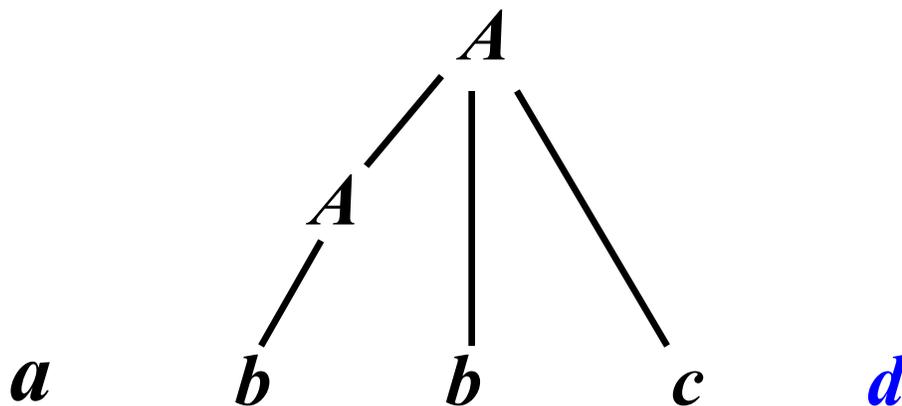
ab (读入 ab)

aA (归约)

$aAbc$ (再读入 bc)

aA (归约)

aAd (再读入 d)





归约(Reduce)



- 每一步，**特定子串**被替换为相匹配的某个**产生式左部的非终结符**
- 最终，把**输入串**归约成文法的**开始符号**

例 $S \rightarrow aABe$

$A \rightarrow Abc / b$

$B \rightarrow d$

输入串: $abbcde$

ab (读入 ab)

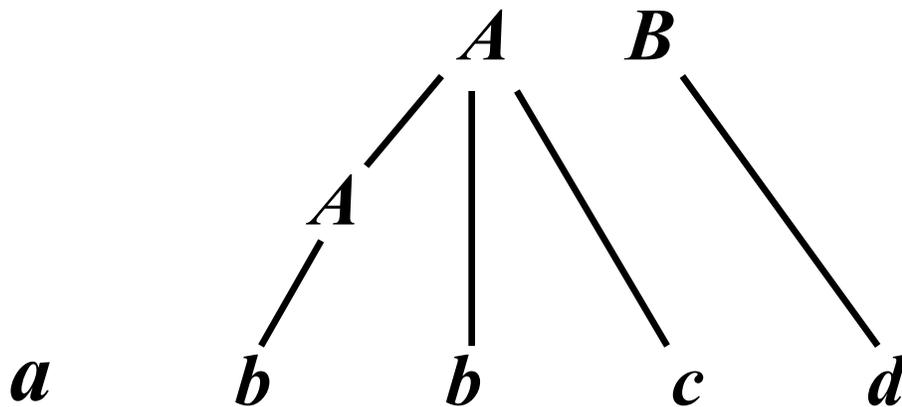
aA (归约)

$aAbc$ (再读入 bc)

aA (归约)

aAd (再读入 d)

aAB (归约)





归约(Reduce)



- 每一步，特定子串被替换为相匹配的某个产生式左部的非终结符
- 最终，把输入串归约成文法的开始符号

例 $S \rightarrow aABe$

$A \rightarrow Abc / b$

$B \rightarrow d$

输入串: $abbcde$

ab (读入 ab)

aA (归约)

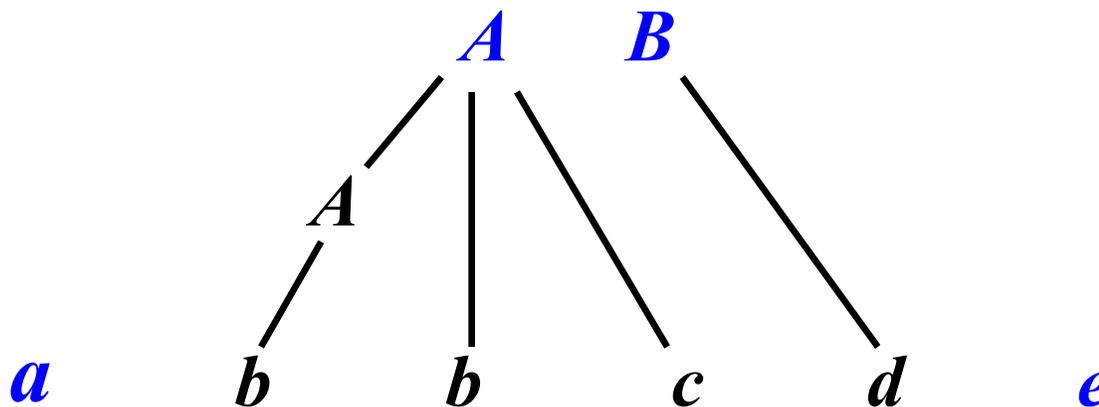
$aAbc$ (再读入 bc)

aA (归约)

aAd (再读入 d)

aAB (归约)

$aABe$ (再读入 e)





归约(Reduce)



- 每一步，**特定子串**被替换为相匹配的某个**产生式左部的非终结符**
- 最终，把**输入串**归约成文法的**开始符号**

例 $S \rightarrow aABe$

$A \rightarrow Abc \mid b$

$B \rightarrow d$

输入串: $abbcde$

ab (读入 ab)

aA (归约)

$aAbc$ (再读入 bc)

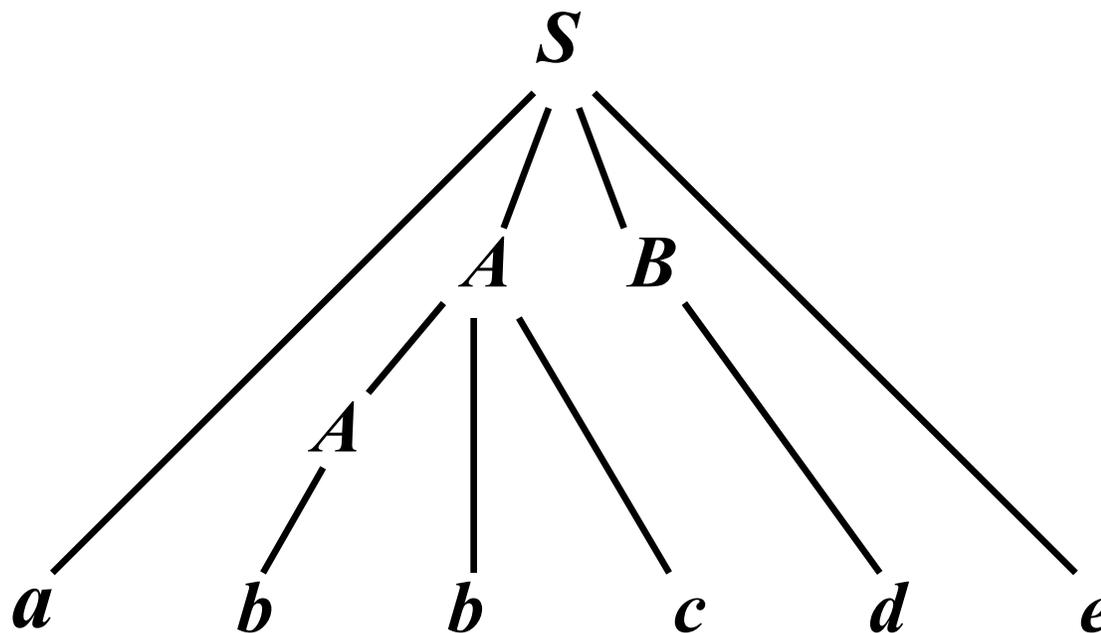
aA (归约)

aAd (再读入 d)

aAB (归约)

$aABe$ (再读入 e)

S (归约)



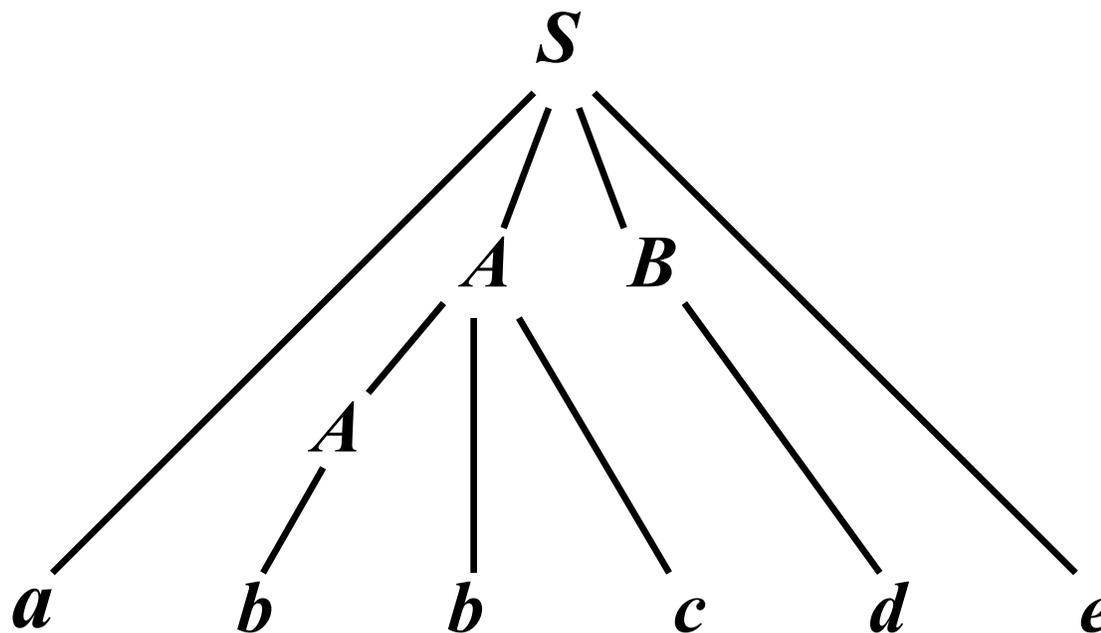


归约(Reduce)



- 每一步，**特定子串**被替换为相匹配的某个**产生式左部的非终结符**
- 最终，把**输入串**归约成文法的**开始符号**
- 归约是最右推导的**逆过程**

例 $S \rightarrow aABe$
 $A \rightarrow Abc \mid b$
 $B \rightarrow d$
 输入串: $abbcde$
 ab (读入 ab)
 $aAbc$ (再读入 bc)
 aAd (再读入 d)
 $aABe$ (再读入 e)
 S (归约)



$$S \Rightarrow_{rm} aABe \Rightarrow_{rm} aAde \Rightarrow_{rm} aAbcde \Rightarrow_{rm} abbcde$$

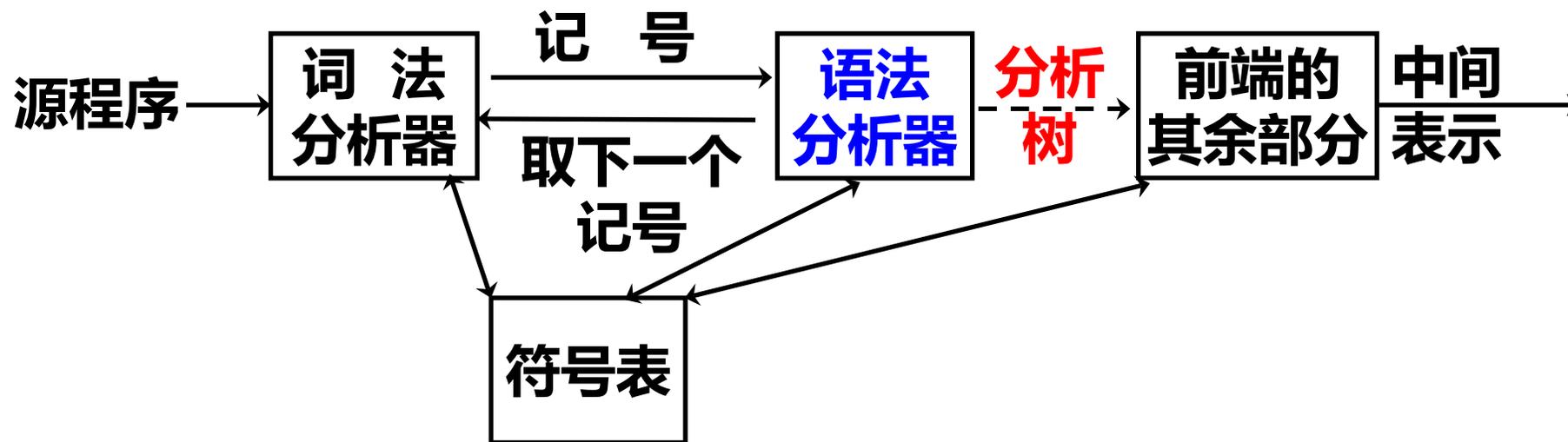


归约(Reduce)



- **需要解决两个问题**

- 在读入串的过程中，如何识别可以归约的子串？
- 在进行归约的时候，选择哪一个产生式？



• 自底向上分析方法

- 归约(右推导的逆过程)
- 句柄(可归约串), 可能不唯一
- 移进-归约分析方法
- 冲突: 移进-归约、归约-归约



• 句型的句柄 (可归约串)

- 该句型中和某产生式右部匹配的子串，并且
- 把它归约成该产生式左部的非终结符，代表了最右推导的逆过程的一步

$$S \rightarrow aABe$$

$$A \rightarrow Abc \mid b$$

$$B \rightarrow d$$

$$S \Rightarrow_{rm} aABe \Rightarrow_{rm} aAde \Rightarrow_{rm} aAbcde \Rightarrow_{rm} abcde$$

- 句柄的右边仅含终结符
- 如果文法二义，那么句柄可能不唯一



例句柄不唯一


$$E \rightarrow E + E \mid E * E \mid (E) \mid \text{id}$$



例句柄不唯一



$$E \rightarrow E + E \mid E * E \mid (E) \mid \text{id}$$

$$\begin{aligned} E &\Rightarrow_{rm} E * E \\ &\Rightarrow_{rm} E * E + E \\ &\Rightarrow_{rm} E * E + \text{id}_3 \\ &\Rightarrow_{rm} E * \text{id}_2 + \text{id}_3 \\ &\Rightarrow_{rm} \text{id}_1 * \text{id}_2 + \text{id}_3 \end{aligned}$$



例句柄不唯一

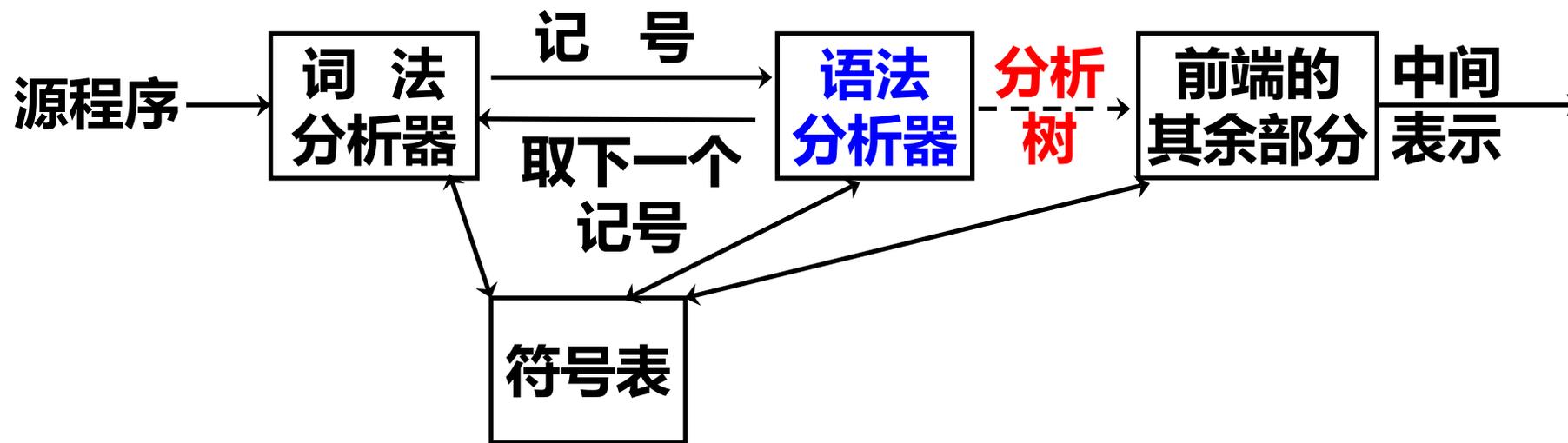


$$E \rightarrow E + E \mid E * E \mid (E) \mid \text{id}$$

$$\begin{aligned}
E &\Rightarrow_{rm} E * E \\
&\Rightarrow_{rm} E * E + E \\
&\Rightarrow_{rm} E * E + \text{id}_3 \\
&\Rightarrow_{rm} E * \text{id}_2 + \text{id}_3 \\
&\Rightarrow_{rm} \text{id}_1 * \text{id}_2 + \text{id}_3
\end{aligned}$$

$$\begin{aligned}
E &\Rightarrow_{rm} E + E \\
&\Rightarrow_{rm} E + \text{id}_3 \\
&\Rightarrow_{rm} E * E + \text{id}_3 \\
&\Rightarrow_{rm} E * \text{id}_2 + \text{id}_3 \\
&\Rightarrow_{rm} \text{id}_1 * \text{id}_2 + \text{id}_3
\end{aligned}$$

在句型 $E * E + \text{id}_3$ 中，句柄不唯一



• 自底向上分析方法

- 归约(右推导的逆过程)
- 句柄(可归约串), 可能不唯一
- 移进-归约分析方法
- 冲突: 移进-归约、归约-归约



• 用栈实现移进-归约分析

- 栈保存已扫描过的文法符号，缓冲区存放还未分析的其余符号
- 移进(shift): 将下一个输入符号放到栈顶，以形成句柄
- 归约(reduce): 将句柄替换为对应的产生式的左部非终结符
- 接受(accept): 分析成功
- 报错(error): 发现语法错误



- **用栈实现移进-归约分析**

- 先通过分析输入串 $id_1 * id_2 + id_3$ 时的动作序列来了解移进-归约分析的工作方式



移进-归约 $id_1 * id_2 + id_3$



栈	输入	动作
\$	$id_1 * id_2 + id_3 \$$	移进
\$ id_1	$* id_2 + id_3 \$$	按 $E \rightarrow id$ 归约
\$ E	$* id_2 + id_3 \$$	移进
\$ $E*$	$id_2 + id_3 \$$	移进
\$ $E*id_2$	$+ id_3 \$$	按 $E \rightarrow id$ 归约
\$ $E*E$	$+ id_3 \$$	



移进-归约 $id_1 * id_2 + id_3$



栈	输入	动作
\$	$id_1 * id_2 + id_3 \$$	移进
\$ id_1	$* id_2 + id_3 \$$	按 $E \rightarrow id$ 归约
\$ E	$* id_2 + id_3 \$$	移进
\$ $E*$	$id_2 + id_3 \$$	移进
\$ $E*id_2$	$+ id_3 \$$	按 $E \rightarrow id$ 归约
\$ $E*E$	$+ id_3 \$$	移进



移进-归约 $id_1 * id_2 + id_3$



栈	输入	动作
\$	$id_1 * id_2 + id_3 \$$	移进
$\$ id_1$	$* id_2 + id_3 \$$	按 $E \rightarrow id$ 归约
$\$ E$	$* id_2 + id_3 \$$	移进
$\$ E *$	$id_2 + id_3 \$$	移进
$\$ E * id_2$	$+ id_3 \$$	按 $E \rightarrow id$ 归约
$\$ E * E$	$+ id_3 \$$	移进
$\$ E * E +$	$id_3 \$$	



移进-归约 $id_1 * id_2 + id_3$



栈	输入	动作
\$	$id_1 * id_2 + id_3 \$$	移进
\$ id_1	$* id_2 + id_3 \$$	按 $E \rightarrow id$ 归约
\$ E	$* id_2 + id_3 \$$	移进
\$ $E*$	$id_2 + id_3 \$$	移进
\$ $E*id_2$	$+ id_3 \$$	按 $E \rightarrow id$ 归约
\$ $E*E$	$+ id_3 \$$	移进
\$ $E*E+$	$id_3 \$$	移进



移进-归约 $id_1 * id_2 + id_3$



栈	输入	动作
\$	$id_1 * id_2 + id_3 \$$	移进
$\$ id_1$	$* id_2 + id_3 \$$	按 $E \rightarrow id$ 归约
$\$ E$	$* id_2 + id_3 \$$	移进
$\$ E *$	$id_2 + id_3 \$$	移进
$\$ E * id_2$	$+ id_3 \$$	按 $E \rightarrow id$ 归约
$\$ E * E$	$+ id_3 \$$	移进
$\$ E * E +$	$id_3 \$$	移进
$\$ E * E + id_3$	\$	



移进-归约 $id_1 * id_2 + id_3$



栈	输入	动作
\$	$id_1 * id_2 + id_3 \$$	移进
\$ id_1	$* id_2 + id_3 \$$	按 $E \rightarrow id$ 归约
\$ E	$* id_2 + id_3 \$$	移进
\$ $E*$	$id_2 + id_3 \$$	移进
\$ $E*id_2$	$+ id_3 \$$	按 $E \rightarrow id$ 归约
\$ $E*E$	$+ id_3 \$$	移进
\$ $E*E+$	$id_3 \$$	移进
\$ $E*E+id_3$	\$	按 $E \rightarrow id$ 归约



移进-归约 $id_1 * id_2 + id_3$



栈	输入	动作
\$	$id_1 * id_2 + id_3 \$$	移进
\$ id_1	$* id_2 + id_3 \$$	按 $E \rightarrow id$ 归约
\$ E	$* id_2 + id_3 \$$	移进
\$ $E*$	$id_2 + id_3 \$$	移进
\$ $E*id_2$	$+ id_3 \$$	按 $E \rightarrow id$ 归约
\$ $E*E$	$+ id_3 \$$	移进
\$ $E*E+$	$id_3 \$$	移进
\$ $E*E+id_3$	\$	按 $E \rightarrow id$ 归约
\$ $E*E+E$	\$	



移进-归约 $id_1 * id_2 + id_3$



栈	输入	动作
\$	$id_1 * id_2 + id_3 \$$	移进
\$ id_1	$* id_2 + id_3 \$$	按 $E \rightarrow id$ 归约
\$ E	$* id_2 + id_3 \$$	移进
\$ $E*$	$id_2 + id_3 \$$	移进
\$ $E*id_2$	$+ id_3 \$$	按 $E \rightarrow id$ 归约
\$ $E*E$	$+ id_3 \$$	移进
\$ $E*E+$	$id_3 \$$	移进
\$ $E*E+id_3$	\$	按 $E \rightarrow id$ 归约
\$ $E*E+E$	\$	按 $E \rightarrow E+E$ 归约



移进-归约 $id_1 * id_2 + id_3$



栈	输入	动作
\$	$id_1 * id_2 + id_3 \$$	移进
$\$ id_1$	$* id_2 + id_3 \$$	按 $E \rightarrow id$ 归约
$\$ E$	$* id_2 + id_3 \$$	移进
$\$ E *$	$id_2 + id_3 \$$	移进
$\$ E * id_2$	$+ id_3 \$$	按 $E \rightarrow id$ 归约
$\$ E * E$	$+ id_3 \$$	移进
$\$ E * E +$	$id_3 \$$	移进
$\$ E * E + id_3$	\$	按 $E \rightarrow id$ 归约
$\$ E * E + E$	\$	按 $E \rightarrow E + E$ 归约
$\$ E * E$	\$	



移进-归约 $id_1 * id_2 + id_3$



栈	输入	动作
\$	$id_1 * id_2 + id_3 \$$	移进
\$ id_1	$* id_2 + id_3 \$$	按 $E \rightarrow id$ 归约
\$ E	$* id_2 + id_3 \$$	移进
\$ $E*$	$id_2 + id_3 \$$	移进
\$ $E*id_2$	$+ id_3 \$$	按 $E \rightarrow id$ 归约
\$ $E*E$	$+ id_3 \$$	移进
\$ $E*E+$	$id_3 \$$	移进
\$ $E*E+id_3$	\$	按 $E \rightarrow id$ 归约
\$ $E*E+E$	\$	按 $E \rightarrow E+E$ 归约
\$ $E*E$	\$	按 $E \rightarrow E*E$ 归约



移进-归约 $id_1 * id_2 + id_3$



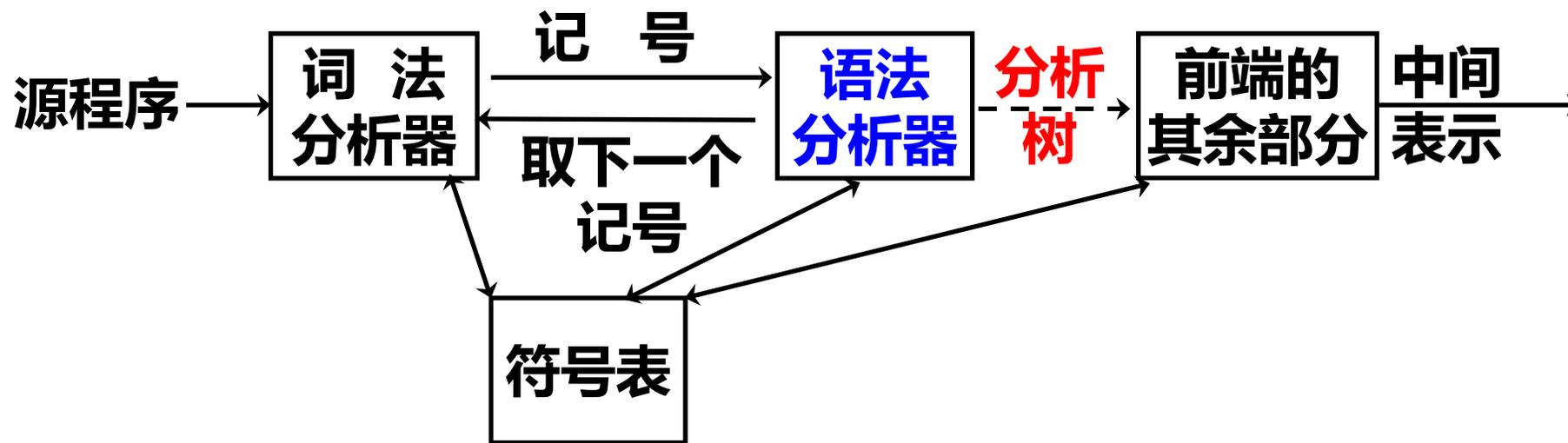
栈	输入	动作
\$	$id_1 * id_2 + id_3 \$$	移进
\$ id_1	$* id_2 + id_3 \$$	按 $E \rightarrow id$ 归约
\$ E	$* id_2 + id_3 \$$	移进
\$ $E*$	$id_2 + id_3 \$$	移进
\$ $E*id_2$	$+ id_3 \$$	按 $E \rightarrow id$ 归约
\$ $E*E$	$+ id_3 \$$	移进
\$ $E*E+$	$id_3 \$$	移进
\$ $E*E+id_3$	\$	按 $E \rightarrow id$ 归约
\$ $E*E+E$	\$	按 $E \rightarrow E+E$ 归约
\$ $E*E$	\$	按 $E \rightarrow E*E$ 归约
\$ E	\$	



移进-归约 $id_1 * id_2 + id_3$



栈	输入	动作
\$	$id_1 * id_2 + id_3 \$$	移进
\$ id_1	$* id_2 + id_3 \$$	按 $E \rightarrow id$ 归约
\$ E	$* id_2 + id_3 \$$	移进
\$ $E*$	$id_2 + id_3 \$$	移进
\$ $E*id_2$	$+ id_3 \$$	按 $E \rightarrow id$ 归约
\$ $E*E$	$+ id_3 \$$	移进
\$ $E*E+$	$id_3 \$$	移进
\$ $E*E+id_3$	\$	按 $E \rightarrow id$ 归约
\$ $E*E+E$	\$	按 $E \rightarrow E+E$ 归约
\$ $E*E$	\$	按 $E \rightarrow E*E$ 归约
\$ E	\$	接受



• 自底向上分析方法

- 归约(右推导的逆过程)
- 句柄(可归约串), 可能不唯一
- 移进-归约分析方法
- 冲突: 移进-归约、归约-归约



例 $stmt \rightarrow$ if $expr$ then $stmt$
| if $expr$ then $stmt$ else $stmt$
| other

如果移进-归约分析器处于格局(configuration)

栈

... if $expr$ then $stmt$

输入

else ... \$

归约?

移进?



归约-归约冲突



例 $stmt \rightarrow id (parameter_list) \mid expr = expr$

$parameter_list \rightarrow parameter_list, parameter \mid parameter$

$parameter \rightarrow id$

$expr \rightarrow id (expr_list) \mid id$

$expr_list \rightarrow expr_list, expr \mid expr$

由 $A(I, J)$ 开始的语句

栈

... id (id

输入

, id)...

归约成 $expr$ 还是 $parameter$?



归约-归约冲突



例 $stmt \rightarrow \text{procid} (parameter_list) \mid expr = expr$
 $parameter_list \rightarrow parameter_list, parameter \mid parameter$
 $parameter \rightarrow id$
 $expr \rightarrow id (expr_list) \mid id$
 $expr_list \rightarrow expr_list, expr \mid expr$

由 $A(I, J)$ 开始的语句 (词法分析查符号表, 区分第一个id)

栈

... **procid** (id

输入

, id)...

需要修改文法中的第一个产生式, 并利用栈中信息



一起努力 打造国产基础软硬件体系!

李诚

国家高性能计算中心(合肥)、信息与计算机国家级实验教学示范中心

计算机科学与技术学院

2024年09月18日